

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

WEST Search History

DATE: Tuesday, June 08, 2004

Hide?	Set Name	Query	Hit Count
		<i>DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>	
<input type="checkbox"/>	L25	L23 and (download\$4 or install\$7)	4
<input type="checkbox"/>	L24	L23 same (download\$4 or install\$7)	0
<input type="checkbox"/>	L23	L22 same associat\$7	14
<input type="checkbox"/>	L22	L20 same refer\$7	99
<input type="checkbox"/>	L21	l14 and l20	13
<input type="checkbox"/>	L20	(stor\$4 near3 program) same (hardware adj configuration)	363
<input type="checkbox"/>	L19	(download\$7 near2 hardware near2 configur\$9)	8
<input type="checkbox"/>	L18	(deploy\$7 near2 hardware near2 configur\$9)	3
<input type="checkbox"/>	L17	l14 and L16	1
<input type="checkbox"/>	L16	(deploy\$7 near2 computer near2 program)	15
<input type="checkbox"/>	L15	l1 and L14	4
<input type="checkbox"/>	L14	l7 or l8 or l9 or l10 or l11 or l12 or L13	3532
<input type="checkbox"/>	L13	713/100.ccls.	667
<input type="checkbox"/>	L12	713/2.ccls.	827
<input type="checkbox"/>	L11	713/1.ccls.	983
<input type="checkbox"/>	L10	714/48.ccls.	589
<input type="checkbox"/>	L9	714/47.ccls.	702
<input type="checkbox"/>	L8	714/46.ccls.	377
<input type="checkbox"/>	L7	714/26.ccls.	169
<input type="checkbox"/>	L6	L1.clm.	0
<input type="checkbox"/>	L5	L1.ab.	11
<input type="checkbox"/>	L4	L1 same ((different or second or another) near2 (computer or system or processor or cpu))	0
<input type="checkbox"/>	L3	L1 same configur\$9	5
<input type="checkbox"/>	L2	L1 same deploy\$7	0
<input type="checkbox"/>	L1	(enabl\$4 or allow\$4) near3 ((proper\$2 or appropriat\$4 or correct\$4) near2 operation) near5 (program\$4 or application or software)	58

END OF SEARCH HISTORY

First Hit☐ **Generate Collection** **Print**

L19: Entry 7 of 8

File: DWPI

Oct 29, 1999

DERWENT-ACC-NO: 2000-028957

DERWENT-WEEK: 200003

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Download procedure for program file, parameter file between terminals connected to communication network - involves downloading program by server in response to hardware configuration information from terminal device for determining specific parameter

Basic Abstract Text (1):

NOVELTY - Terminal device (3) downloads hardware configuration information to server (1). The server downloads program to terminal device in response to hardware configuration information for determining specific parameter. The server downloads parameter to terminal device, determined by the program downloaded by server.

Basic Abstract Text (3):

ADVANTAGE - Since server is updated with hardware configuration information of terminal device, only corresponding program and parameter file is downloaded to terminal device. Since files specific to hardware configuration is downloaded, communication time and cost are reduced, when program as file corresponding to different hardware. Prevents downloading the program of same version or revision by comparing the parameter is downloaded. Avoids updation of database by hardware modification of terminal device, simplifies maintenance by avoiding database for component information. DESCRIPTION OF DRAWING(S) - The figure shows profile explanatory diagram of download process by comparing component information, for program, parameter file to terminal device. (1) Server; (3) Terminal device.

First Hit☐ **Generate Collection** **Print**

L19: Entry 7 of 8

File: DWPI

Oct 29, 1999

DERWENT-ACC-NO: 2000-028957

DERWENT-WEEK: 200003

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Download procedure for program file, parameter file between terminals connected to communication network - involves downloading program by server in response to hardware configuration information from terminal device for determining specific parameter

PATENT-ASSIGNEE:

ASSIGNEE

CODE

DAINI DENDEN KK

DAINN

PRIORITY-DATA: 1998JP-0117863 (April 13, 1998)

Search Selected**Search ALL****Clear**

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
<input type="checkbox"/> <u>JP 11296453 A</u>	October 29, 1999		007	G06F013/00

APPLICATION-DATA:

PUB-NO	APPL-DATE	APPL-NO	DESCRIPTOR
JP 11296453A	April 13, 1998	1998JP-0117863	

INT-CL (IPC): G06 F 9/445; G06 F 13/00

ABSTRACTED-PUB-NO: JP 11296453A

BASIC-ABSTRACT:

NOVELTY - Terminal device (3) downloads hardware configuration information to server (1). The server downloads program to terminal device in response to hardware configuration information for determining specific parameter. The server downloads parameter to terminal device, determined by the program downloaded by server.

USE - For downloading program file, parameter file to terminals connected to communication network e.g. internet.

ADVANTAGE - Since server is updated with hardware configuration information of terminal device, only corresponding program and parameter file is downloaded to terminal device. Since files specific to hardware configuration is downloaded, communication time and cost are reduced, when program as file corresponding to different hardware. Prevents downloading the program of same version or revision by comparing the parameter is downloaded. Avoids updation of database by hardware modification of terminal device, simplifies maintenance by avoiding database for component information. DESCRIPTION OF DRAWING(S) - The figure shows profile

explanatory diagram of download process by comparing component information, for program, parameter file to terminal device. (1) Server; (3) Terminal device.

CHOSEN-DRAWING: Dwg.5/8

TITLE-TERMS: PROCEDURE PROGRAM FILE PARAMETER FILE TERMINAL CONNECT COMMUNICATE
NETWORK PROGRAM SERVE RESPOND HARDWARE CONFIGURATION INFORMATION TERMINAL DEVICE
DETERMINE SPECIFIC PARAMETER

DERWENT-CLASS: T01

EPI-CODES: T01-F01B; T01-F05B; T01-H;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N2000-021981

First Hit**End of Result Set**☐ **Generate Collection** **Print**

L19: Entry 8 of 8

File: DWPI

Oct 2, 2003

DERWENT-ACC-NO: 1999-214447

DERWENT-WEEK: 200372

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Computer implemented method for generating graphical codes from hardware implementation

Basic Abstract Text (2):

DETAILED DESCRIPTION - A graphical program portion is created which is compiled into machine code for execution by CPU, and second portion of program comprising conversion of hardware description into net list, by which the programmable hardware element is configured. The net list format is compiled into hardware program file, and the hardware program file is downloaded to the programmable hardware element to configure the hardware element.

Equivalent Abstract Text (2):

DETAILED DESCRIPTION - A graphical program portion is created which is compiled into machine code for execution by CPU, and second portion of program comprising conversion of hardware description into net list, by which the programmable hardware element is configured. The net list format is compiled into hardware program file, and the hardware program file is downloaded to the programmable hardware element to configure the hardware element.

Equivalent Abstract Text (7):

DETAILED DESCRIPTION - A graphical program portion is created which is compiled into machine code for execution by CPU, and second portion of program comprising conversion of hardware description into net list, by which the programmable hardware element is configured. The net list format is compiled into hardware program file, and the hardware program file is downloaded to the programmable hardware element to configure the hardware element.

Equivalent Abstract Text (12):

DETAILED DESCRIPTION - A graphical program portion is created which is compiled into machine code for execution by CPU, and second portion of program comprising conversion of hardware description into net list, by which the programmable hardware element is configured. The net list format is compiled into hardware program file, and the hardware program file is downloaded to the programmable hardware element to configure the hardware element.

End of Result Set

☐ Generate Collection ☐ Print

Feb 28, 1995

TITLE: Data processing system and method having selectable scheduler

It should therefore be apparent that operating system and data processing system efficiency, portability, and utility may be greatly enhanced by providing an operating system kernel which may utilize multiple diverse schedulers in order to optimize allocation of system resources and allow correct operation of different types of applications on data processing systems which include that operating system kernel.

http://westbrs:9000/bin/cgi-bin/accum_query.pl?MODE=%20%20%20%20Display%20%20%20%20... 6/8/04

First Hit

Generate Collection

Print

L18: Entry 2 of 3

File: EPAB

Feb 27, 2003

DOCUMENT-IDENTIFIER: WO 3017149 A2

TITLE: RECONFIGURABLE MEASUREMENT SYSTEM UTILIZING A PROGRAMMABLE HARDWARE ELEMENT
AND FIXED HARDWARE RESOURCESAbstract Text (1):

A system and method for configuring a device to perform a function, where the device includes a programmable hardware element and one or more fixed hardware resources. A program is stored which represents the function. A hardware configuration program is generated based on the program, specifying a configuration for the programmable hardware element that implements the function, and usage of the fixed hardware resources by the programmable hardware element in performing the function. A deployment program deploys the hardware configuration program onto the programmable hardware element, where, after deployment, the device is operable to perform the function, where the programmable hardware element directly performs a first portion of the function, and the programmable hardware element invokes the fixed hardware resources to perform a second portion of the function. An optional measurement module couples to the device and performs signal conditioning and/or conversion logic on an acquired signal for the device.

First Hit Fwd Refs



Generate Collection

Print

L19: Entry 1 of 8

File: USPT

Jun 18, 2002

DOCUMENT-IDENTIFIER: US 6408074 B1

TITLE: Hardware architecture for a configurable cipher device

Detailed Description Text (2):

Referring to FIG. 1, there is shown a block diagram of configurable architecture based cipher device (hereinafter "cipher device") 100 of the present invention coupled to information source 300 via Input/Output (I/O) path 200. Cipher device 100 performs cryptographic processing on information it receives from information source 300 via I/O path 200 and outputs the processed information to information source 300. It should be noted that information source 300 is shown for illustrative purposes and does not form part of the cipher device (100) of the present invention. Cipher device 100 is flexible in that it comprises configurable hardware that allows it to perform different types of cryptographic algorithms based on configurations downloaded onto such hardware. Configurable hardware is a device or a plurality of devices whose individual construction and/or collective arrangement are controlled by configurations downloaded onto such hardware. The configurations are commands or instructions that can take the form of information represented by digital or analog signals. Cipher device 100 performs a plurality of cryptographic algorithms simultaneously. Cipher device 100 is efficient in terms of speed of execution, power consumption, physical size and cost of manufacture because its configurable hardware is tailored to cryptographic processing thus significantly reducing the existence of extraneous circuits within the device.

Detailed Description Text (8):

Configurable cipher block 106 is configurable hardware whose internal circuitry is arranged by downloading configurations onto such hardware. The configurations originate from any storage device or system (e.g., information source 300) that is capable of downloading configuration commands (i.e., instructions, typically in digital form) onto configurable hardware. Examples of such hardware are FPLAs and ASICs. In particular, Configurable cipher block 106 comprises a plurality of configurable hardware units called algorithm blocks which can be configured to execute various types of cryptographic algorithms (encryption and/or decryption). The cryptographic algorithms are executed by the algorithm blocks which perform cryptographic operations on information provided via path 116 and transfer the result onto path 118. Thus, Configurable cipher block can be configured for different modes (e.g., encryption, decryption) of operation. One example of a mode of operation is a "no cipher" mode in which information (on path 116) associated with said mode is simply passed through to path 118 without being processed. When a particular algorithm block has finished performing its algorithm, it notifies Control circuit 104 of that event with a control signal via path 114. Control circuit 104 detects the control signal and generates status information and/or control signals which are transferred to I/O interface circuit 102. Information on path 116 are arranged and stored in I/O interface circuit 102 as groups of information or blocks of information which are to be processed in accordance with the command information and/or control signals associated with said information. The information is similarly arranged on path 118. More than one block of information can be transferred to configurable cipher block 106 at any one time. For example, a particular block of information can be transferred to a particular algorithm block within configurable cipher block 106. The flow of information associated with a particular block of information is referred to as an information

stream. Thus, the cipher device of the present invention can process more than one information stream simultaneously. One example is where each information stream is associated with a communication channel of a communication system and each channel generates command information and control signals that dictate the type of processing (e.g. encryption, decryption) to be performed on its block of information. Command information, status information and control signals have a "tag" or identifier that indicates the block of information with which they are associated. In the example of a communication system, the "tag" may be a channel number that identifies the particular communication channel with which the block of information is associated. Other well known techniques for associating command information (or status information) and control signals to particular blocks of information can also be used.

☐ Generate Collection ☐ Print

May 22, 2001

TITLE: Method and apparatus for adaptable digital protocol processing

16. The network processor of claim 1, wherein the computer system is adapted to configure the configurable logic into the analysis configuration by downloading into the configurable logic at least one configuration file associated with the analysis configuration, and wherein the computer system is adapted to configure the configurable logic into the processing configuration by downloading into the adaptable hardware device at least one configuration file associated with the processing configuration.

[First Hit](#) [Fwd Refs](#)**End of Result Set**☐ [Generate Collection](#) [Print](#)

L17: Entry 1 of 1

File: USPT

Jun 12, 2001

DOCUMENT-IDENTIFIER: US 6247145 B1

TITLE: Automated reliability and maintainability process

Detailed Description Text (21):

Data is normally collected on the Form 2581/2 and reviewed by the maintenance supervisor. After being reviewed, the data is then entered in the SEDS database system. This can be accomplished by typing the data directly into the SEDS database, or by using a data collection program that is deployed on a personal computer in the maintenance area. Where the data is collected using deployed program, the data is committed to the main SEDS database the maintenance personnel can no longer modify it.

Current US Cross Reference Classification (1):714/47

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L15: Entry 1 of 4

File: USPT

Dec 9, 2003

DOCUMENT-IDENTIFIER: US 6662316 B1

TITLE: Electronic apparatus having improved diagnostic interface

Detailed Description Text (35):

Within current Intel-based personal computer architectures, the advantage of communicating with a subsystem, such as microcontroller 270 using the output SPKR is that although special programming is required in the BIOS to enable this, such programming need only be carried out once. The output SPKR is a so called 'legacy' feature the programming interface to which does not change with successive generations of chipset products and whose correct operation is required in order to allow forward compatibility of the programs that run on the computer. In consequence, this BIOS code that enables communication with microcontroller 270 via output SPKR will not normally need to be updated for new generations of chipset products 160.

Current US Original Classification (1):714/46

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L15: Entry 2 of 4

File: USPT

Feb 13, 2001

DOCUMENT-IDENTIFIER: US 6189050 B1

TITLE: Method and apparatus for adding or removing devices from a computer system without restarting

Detailed Description Text (40):

Alternatively, in real-time systems where applications are the highest priority processes, instead of running the thread process 502 at a higher priority than application processes, a code semaphore system is employed that ensures that once step 610 is executed and returns an indication that a device is removed, steps 611 and 612 are executed before any other instructions in the system. This ensures correct application operation while allowing the thread process 502 to run at a lower priority than application processes.

Current US Cross Reference Classification (10):713/1

First Hit☐ **Generate Collection** **Print**

L5: Entry 10 of 11

File: DWPI

Nov 5, 1999

DERWENT-ACC-NO: 2000-044452

DERWENT-WEEK: 200004

COPYRIGHT 2004 DERWENT INFORMATION LTD

TITLE: Installation log management method for automatic installation of revision of application program in LSS - involves releasing and updating log management information from storing medium after application information stored in memory device are updated based on comparison result of both information

PATENT-ASSIGNEE:

ASSIGNEE

CODE

NEC CORP

NIDE

PRIORITY-DATA: 1998JP-0121783 (April 15, 1998)

Search Selected**Search ALL****Clear**

PATENT-FAMILY:

PUB-NO

PUB-DATE

LANGUAGE

PAGES

MAIN-IPC

JP 11305999 A

November 5, 1999

010

G06F009/06

APPLICATION-DATA:

PUB-NO

APPL-DATE

APPL-NO

DESCRIPTOR

JP 11305999A

April 15, 1998

1998JP-0121783

INT-CL (IPC): G06 F 9/06

ABSTRACTED-PUB-NO: JP 11305999A

BASIC-ABSTRACT:

NOVELTY - Log management information are released from a storing medium (4) and updated after application information for every front version stored in a memory device (2) are updated based on the comparison result of both information during installation and non-installation operation conditions. DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for an installation log management system.

USE - For automatic installation of revision of application program in LSS.

ADVANTAGE - Generates installation management information in short period of time, thus time of installation operation of application program to LSS is also shortened. Enables efficient and correct installation operation of application program. Also enables efficient and correct installation of system parameter. DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the installation log management system which applies the installation log management method. (2) Memory device; (4) Storing medium.

CHOSEN-DRAWING: Dwg.1/12

TITLE-TERMS: INSTALLATION LOG MANAGEMENT METHOD AUTOMATIC INSTALLATION REVISED
APPLY PROGRAM RELEASE UPDATE LOG MANAGEMENT INFORMATION STORAGE MEDIUM AFTER APPLY
INFORMATION STORAGE MEMORY DEVICE UPDATE BASED COMPARE RESULT INFORMATION

DERWENT-CLASS: T01

EPI-CODES: T01-F06; T01-J20B;

SECONDARY-ACC-NO:

Non-CPI Secondary Accession Numbers: N2000-034032

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L3: Entry 4 of 5

File: USPT

Feb 22, 2000

DOCUMENT-IDENTIFIER: US 6029258 A

TITLE: Method and system for trouble shooting and correcting computer software problems

Brief Summary Text (5):

Unfortunately, computer software applications often malfunction for a number of reasons. Often a software application user fails to properly install a particular software application or component of a particular software application. Often the user's computer is not configured properly to allow proper operation of a particular software application, and inevitably, some software applications contain problems discovered by the providers of those software applications after they are placed on the market and installed by users. Consequently, such problems typically lead to time consuming, expensive, and annoying downtime of computers while users attempt to ascertain the reason for the problem and attempt to effect a correction.

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L23: Entry 6 of 14

File: USPT

Jul 18, 1995

DOCUMENT-IDENTIFIER: US 5434963 A

TITLE: Method and system of help-information control method and system

Detailed Description Text (3):

FIG. 3 is a block diagram showing the hardware configuration of a word-processor, which is an embodiment of the present invention. In the figure, reference numeral 1 is a central processing unit (hereinafter called the CPU). The CPU executes document-editing programs and controls peripheral equipment. Numeral 2 is a main memory (hereinafter called the MM) wherein are stored the document-editing programs and their associated data. The programs to be executed by the CPU include resident and non-resident programs. The resident programs always stay in the MM 2. The non-resident programs are usually stored in a hard disk drive (hereinafter called the HD) 13 and are rolled into the MM 2 or rolled out of it and back to the HD 13 as needed. Numeral 3 is a video random access memory (hereinafter called the VRAM). The contents of the VRAM 3 comprise a display part 3a and a non-display part 3b. The display part 3a is displayed on a cathode ray tube (hereinafter called the CRT) 7. The non-display part 3b provides a temporary save area that accommodates data to be displayed on the CRT 7.

[First Hit](#) [Fwd Refs](#)

Generate Collection

Print

L23: Entry 4 of 14

File: USPT

May 26, 1998

DOCUMENT-IDENTIFIER: US 5758156 A

TITLE: Method and apparatus of testing programs

Detailed Description Text (55):

The correcting section 46 refers to the first correlation table showing the correlation between device names in the catalog data previously stored in the first correlation table storage 38 and the names of programs for testing those devices, and invalidates command lines in the reference catalog data, associated with the hardware configuration of this system, other than those associated with programs registered in the non-coincidence table.